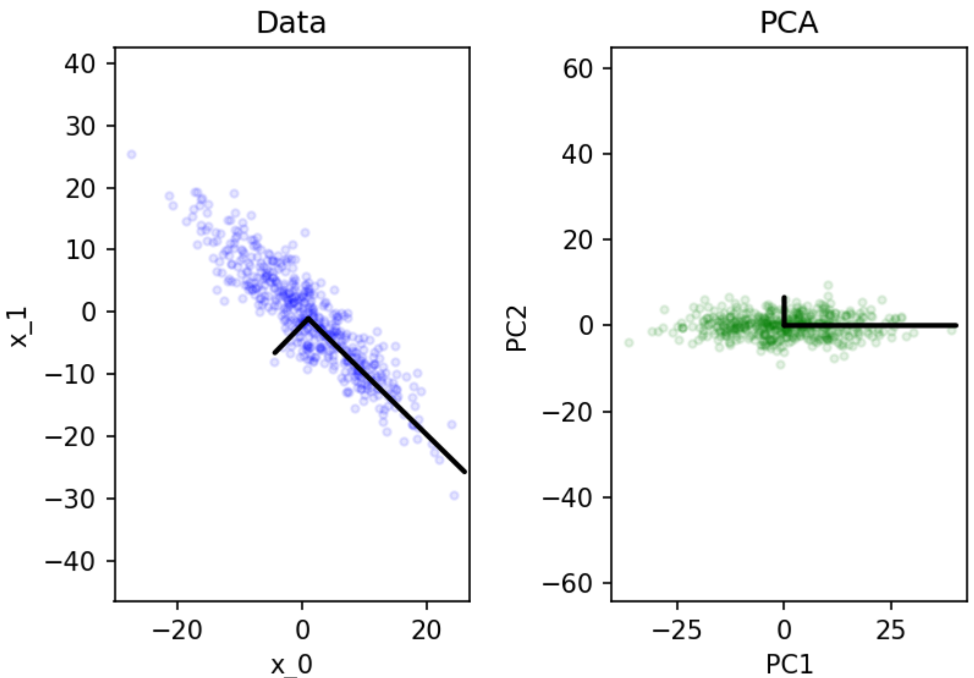
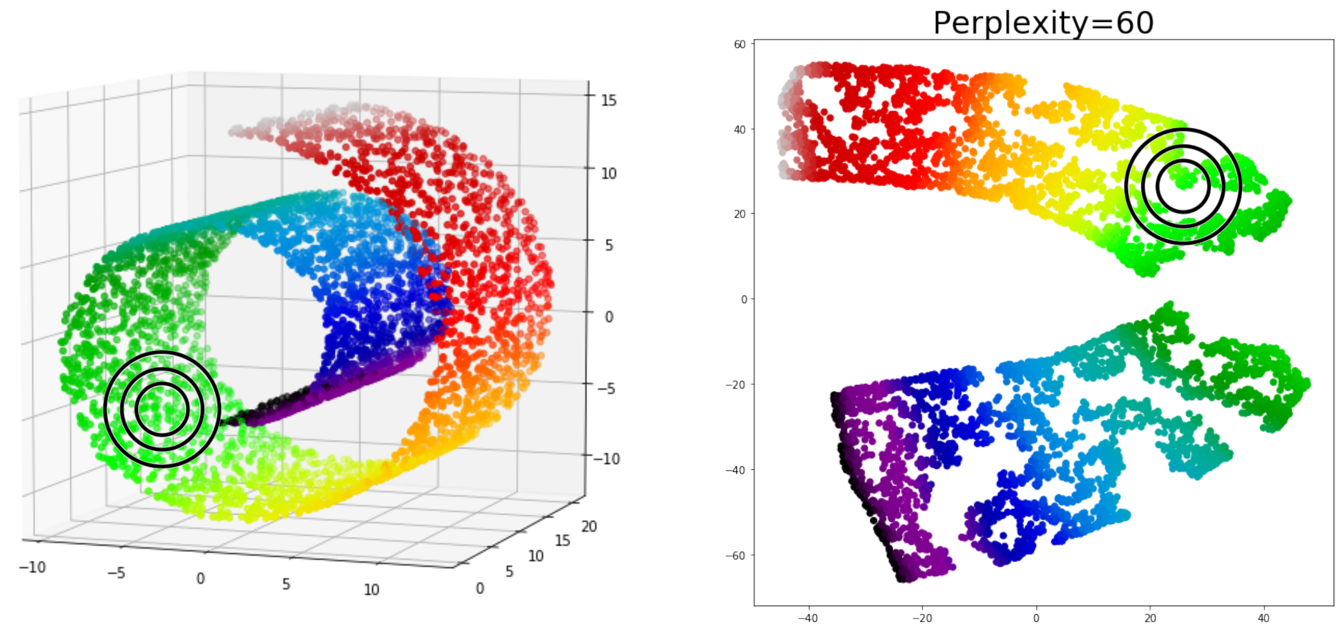
Dimensionality reduction

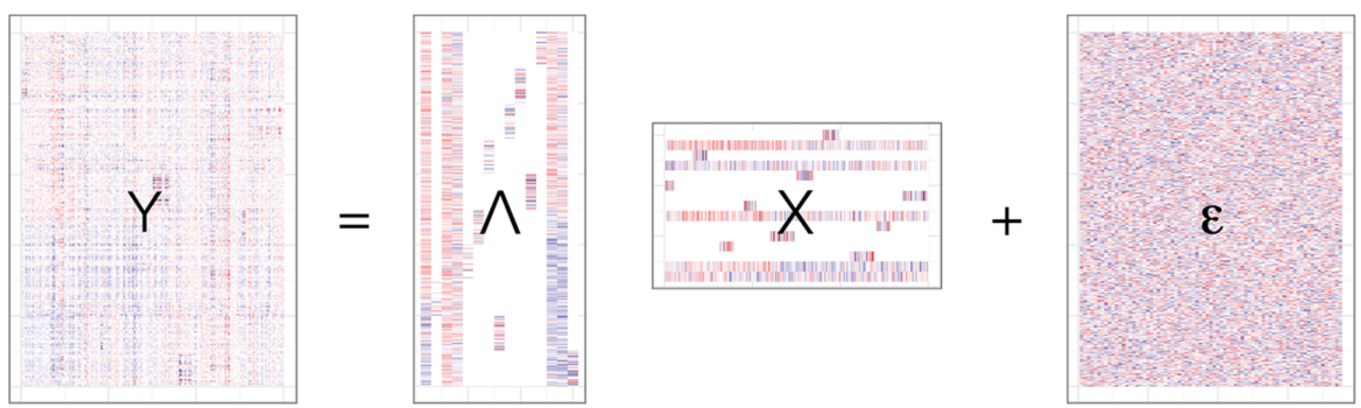
Principal Components Analysis

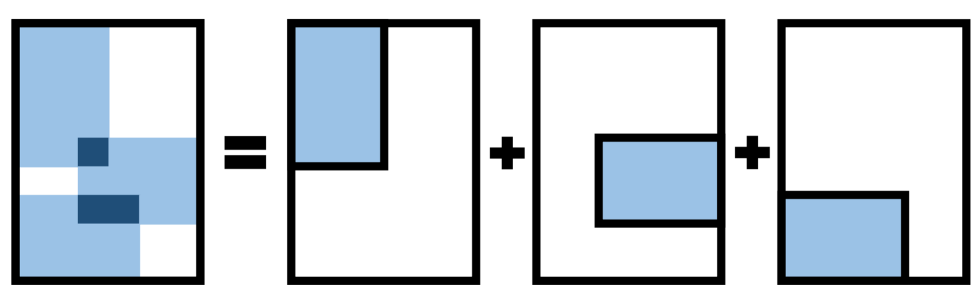


t-Stochastic Neighbour Embedding



Sparse Factor Analysis





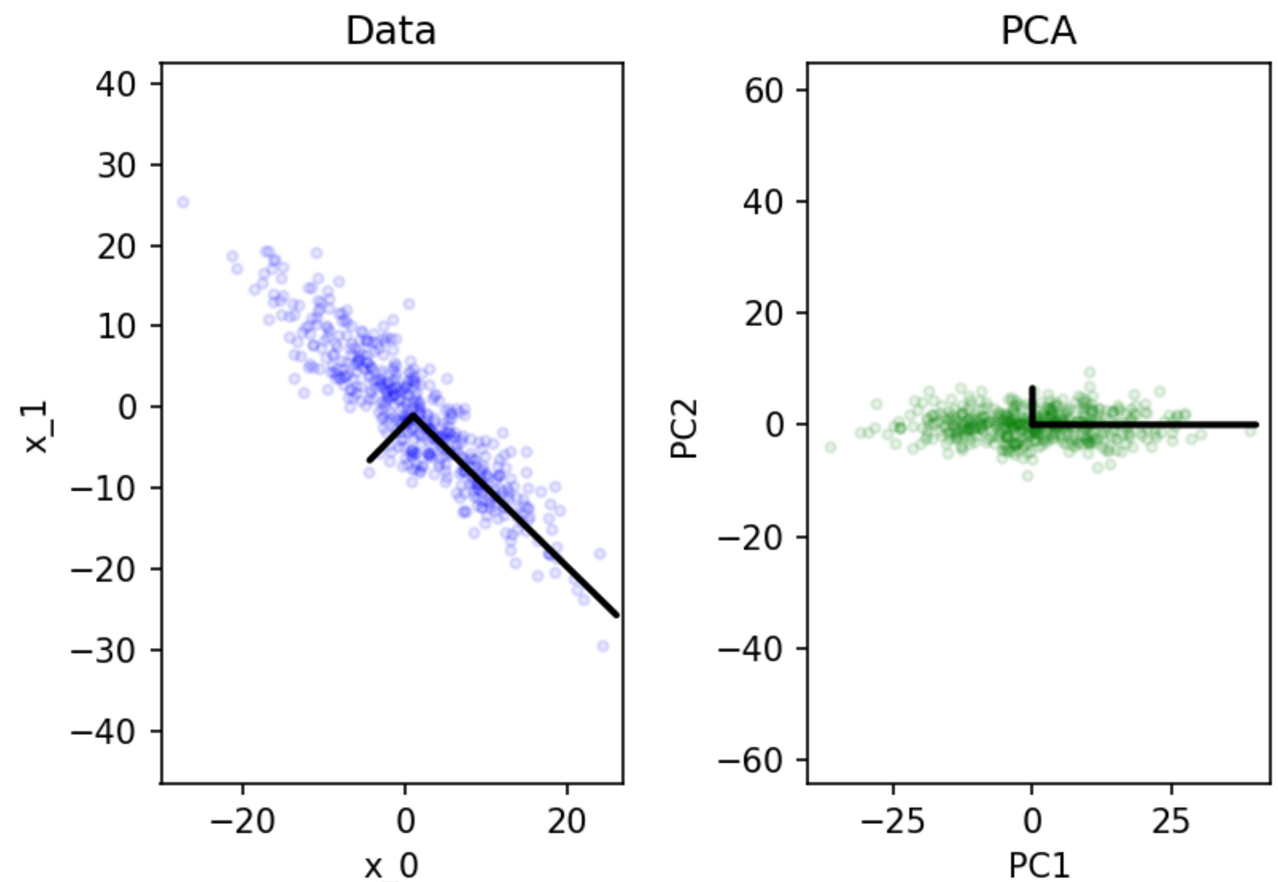
Kath Nicholls

Aims of dimensionality reduction

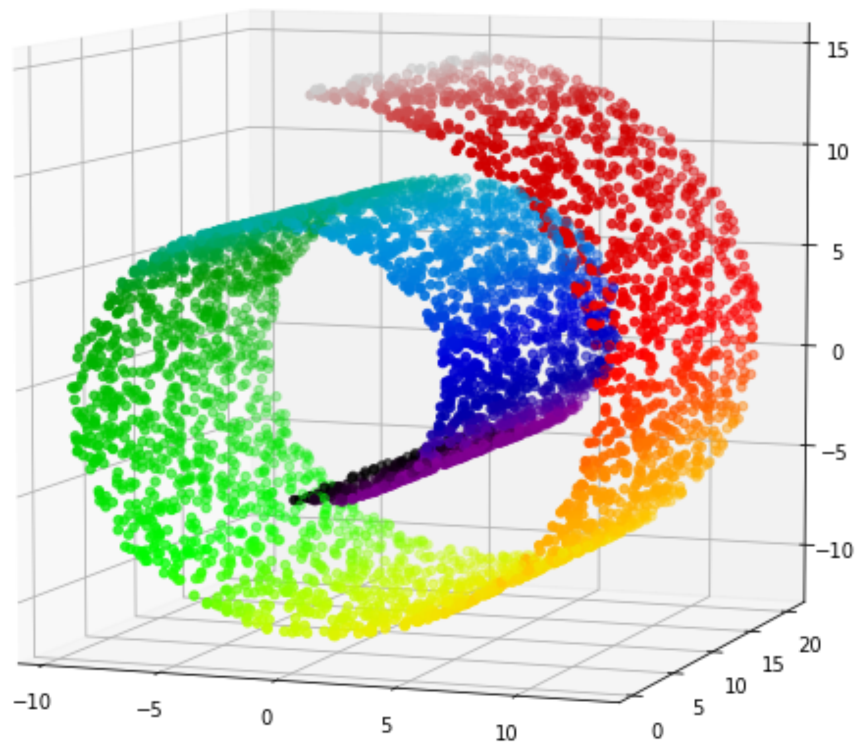
- Produce mapping to reduce dimensions without losing structure
- Visualise high-dimensional data
- Identify important variables
- Cluster data points (e.g. to check for a batch effect)

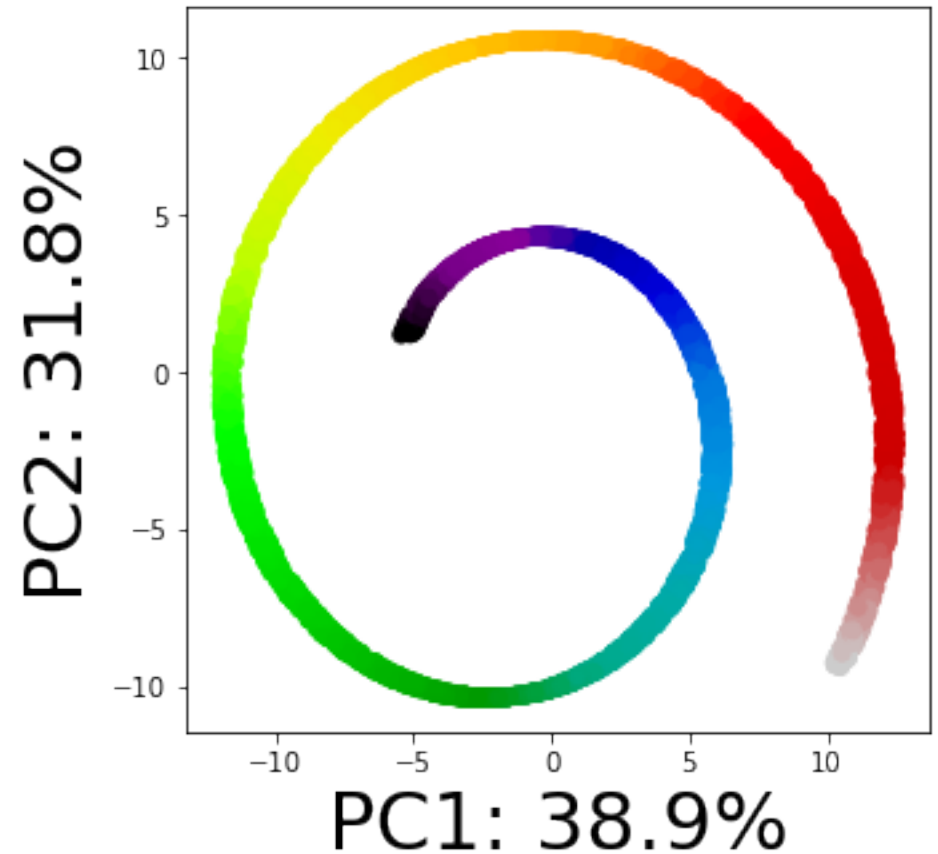
Principal Components Analysis

- Directions of most variance
  - Linear
  - Orthogonal
- Explicit mapping
- Easy to change k
- Calculate using SVD or eigen decomposition



Linearity of PCA





- PCA won't find non-linear directions of variance

PCA via eigen decomposition

- X has n rows (samples) and p columns (variables)
- $C = \frac{1}{n-1} X^T X$ **is $p \times p$ covariance matrix**
- Eigen decomposition is $C = VLV^T$

$$C = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_p \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_p \end{bmatrix} \begin{bmatrix} — & v_1^T & — \\ — & v_2^T & — \\ & \vdots & \\ — & v_p^T & — \end{bmatrix} \qquad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$$

- $X \mapsto XV$ transforms to rotated p-dimensional space
- $X \mapsto XV_{1:k}$ transforms to *reduced* k-dimensional space
- XV_j is called jth *principal component*

PCA via singular value decomposition

- Singular value decomposition of X is

$$X = \underbrace{\begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_n \\ | & | & & | \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \lambda_1^2 & & & & 0 & \cdots & 0 \\ & \lambda_2^2 & & & \vdots & & \vdots \\ & & \ddots & & \vdots & & \vdots \\ & & & \lambda_n^2 & 0 & \cdots & 0 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} — & v_1^T & — \\ — & v_2^T & — \\ & \vdots & \\ & \vdots & \\ & \vdots & \\ — & v_p^T & — \end{bmatrix}}_{V^T}$$

Eigenvectors of XX^T

Eigenvectors of X^TX

- Linked to eigen decomposition

- $XV_j = U_jD_j$ is jth principal component

Reconstructing X

- Recall $X = UDV^T$
- $V_{1:k}V_{1:k}^T \approx I$ (exact for k=p since V orthogonal)

$$X \approx XV_{1:k}V_{1:k}^T$$

- Link to factor analysis:

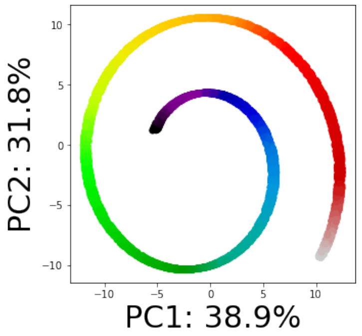
$$X = F\Lambda + \epsilon$$

Principal Components Analysis

- Directions of most variance

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$$

  - Linear



  - Orthogonal
- Explicit mapping

$$X \mapsto XV_{1:k}$$

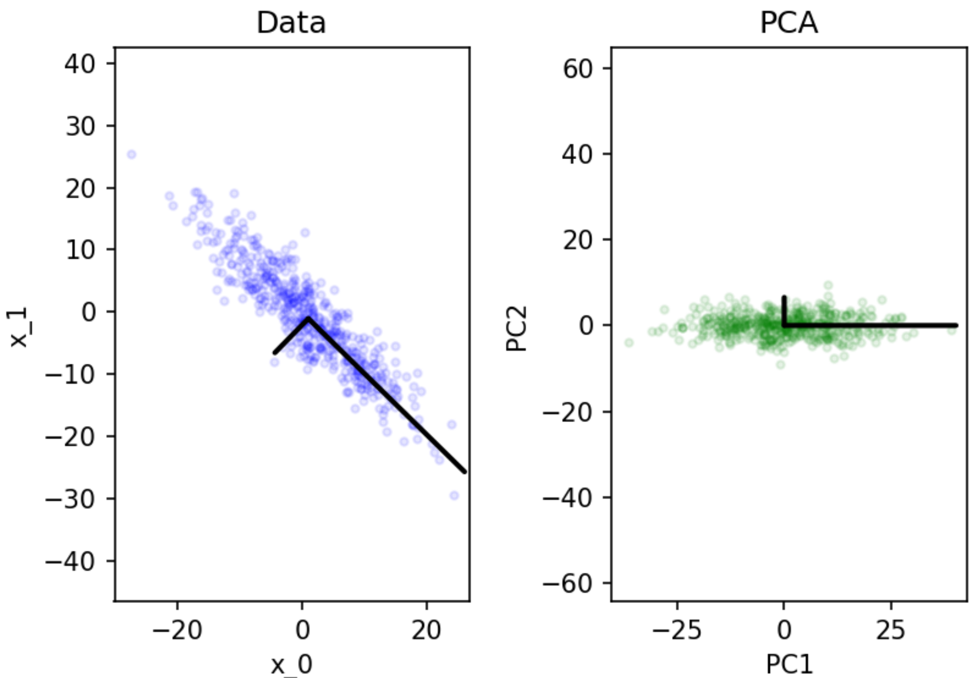
- Easy to change k

$$X \mapsto XV_{1:k}$$

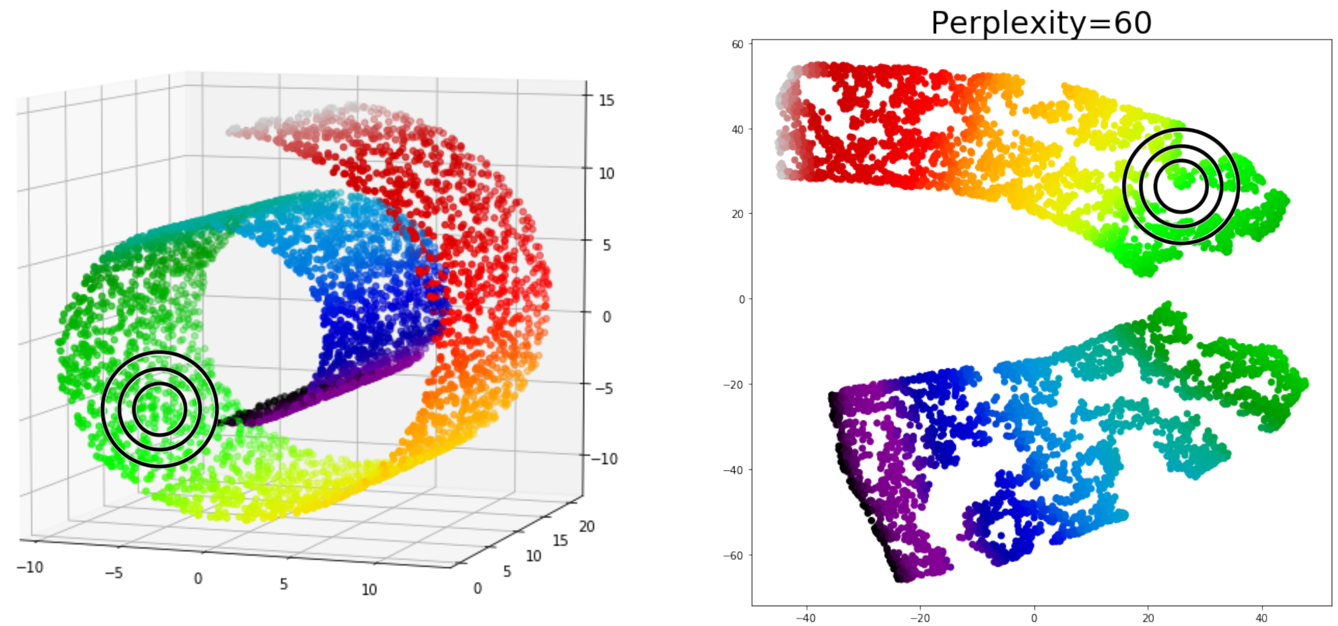
- Calculate using SVD or eigen decomposition

Dimensionality reduction

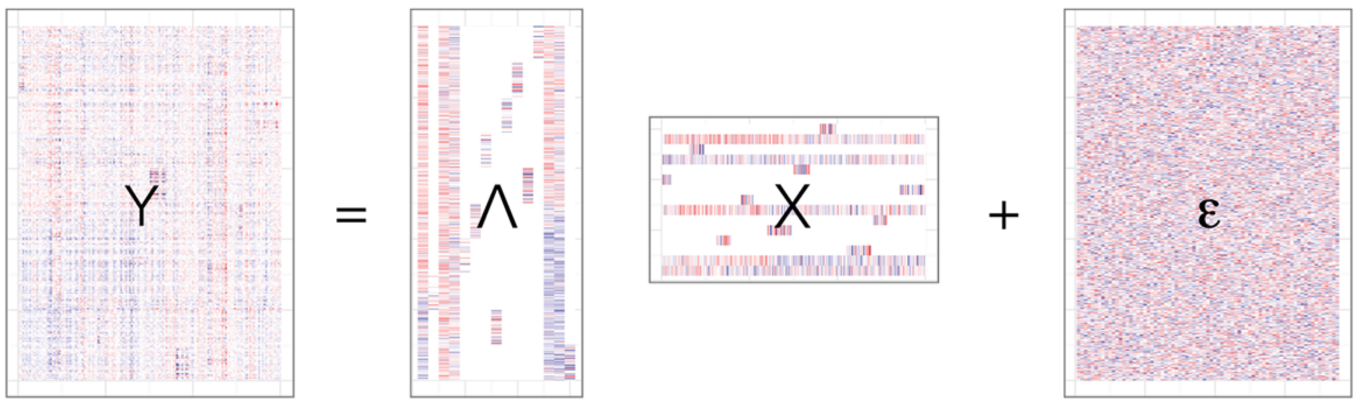
Principal Components Analysis

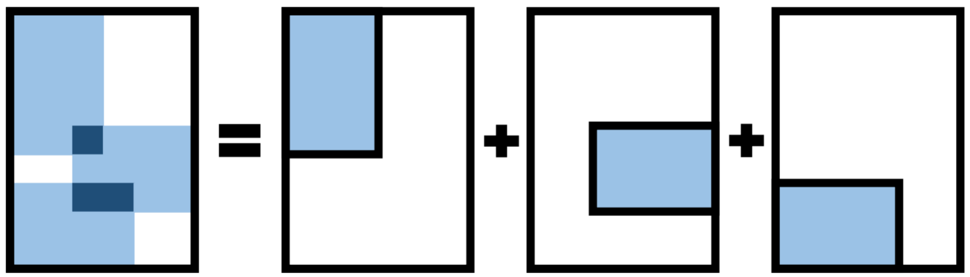


t-Stochastic Neighbour Embedding



Sparse Factor Analysis

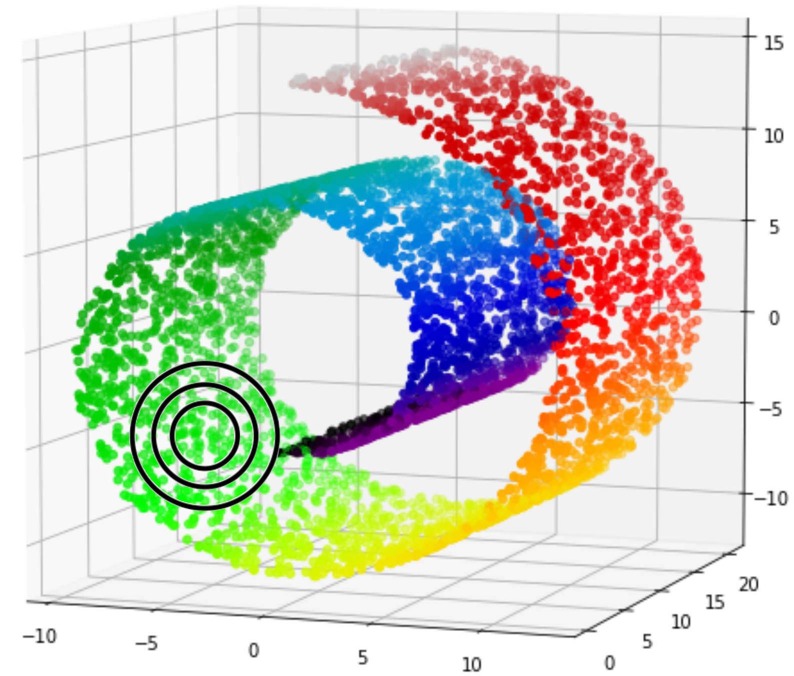


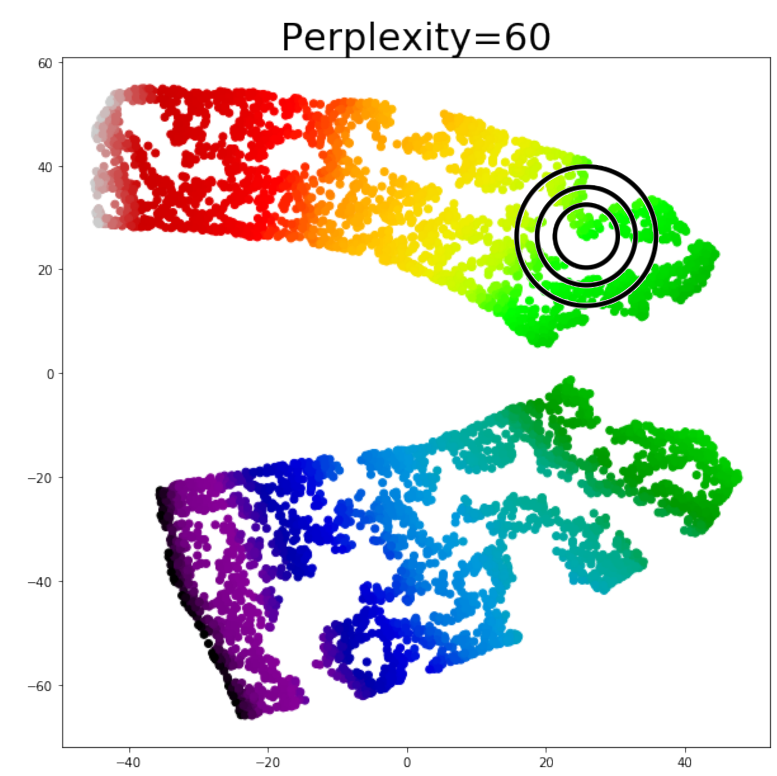


Kath Nicholls

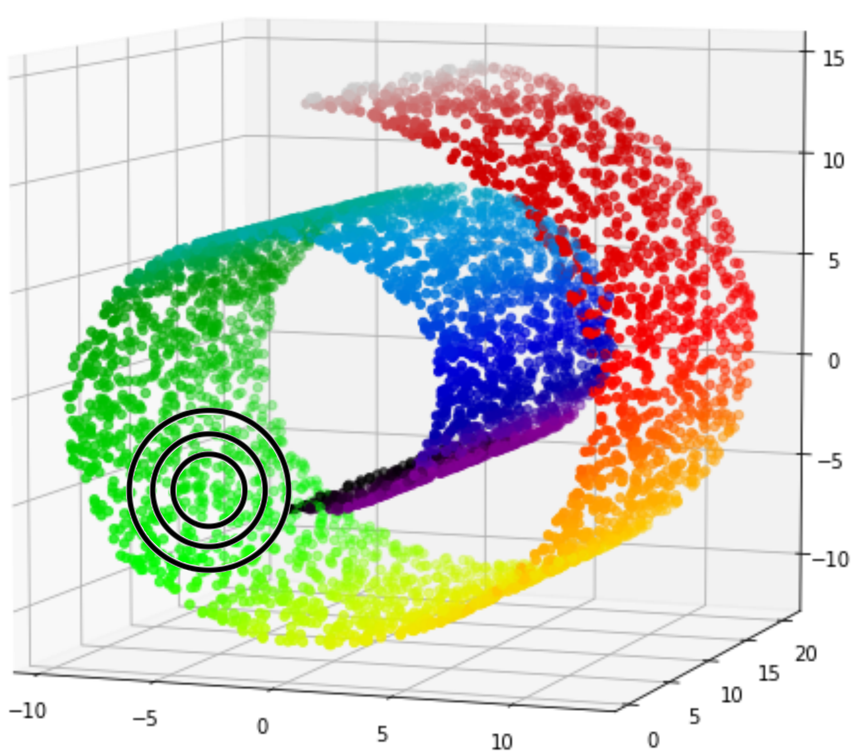
t-distributed Stochastic Neighbour Embedding

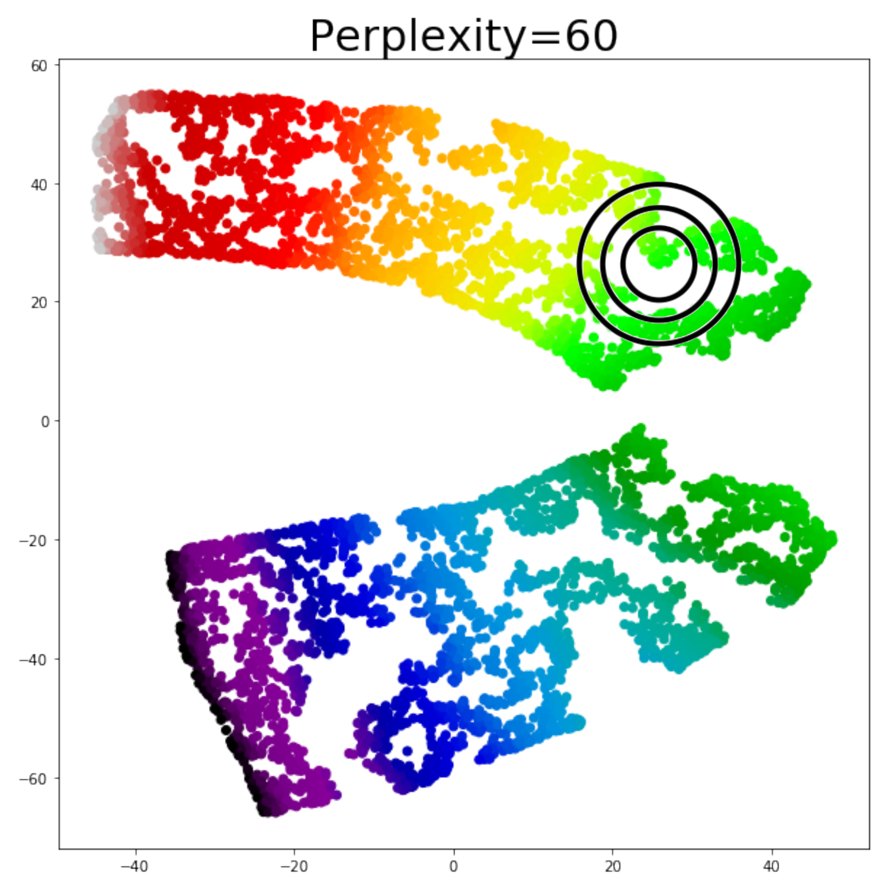
- Preserves local similarity
- Stochastic
- Hyperparameters
    - Perplexity
- No explicit mapping





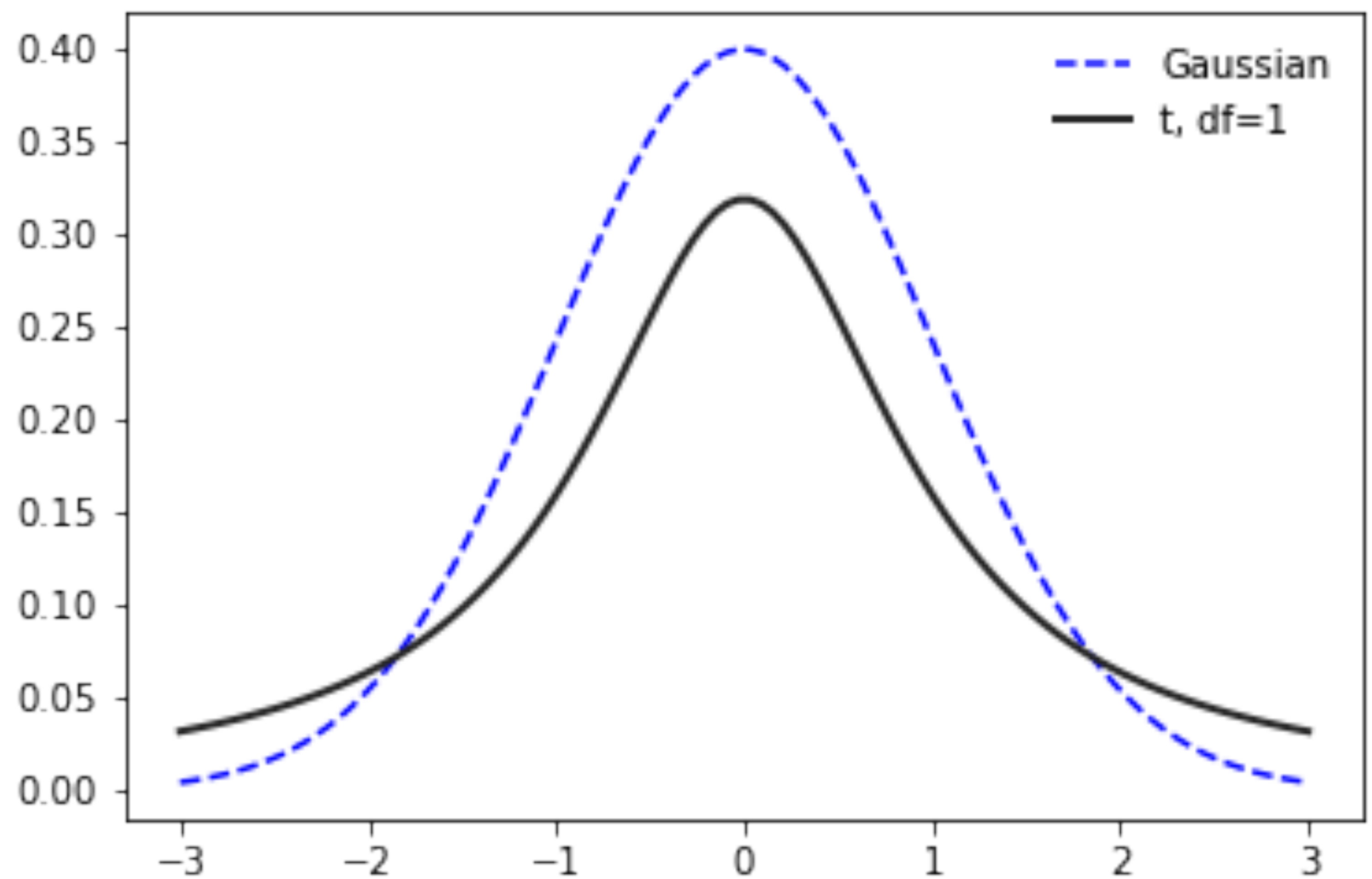
t-SNE preserves local similarity

- $x_1, \ldots, x_n$ in high dimensional space
- Map to points $y_1, \ldots, y_n$ in low dimensional space
- Gaussian centred at x_i gives same points as Student t-distribution centred at y_i
- 



Why Student t-distribution for lower-dimensional points?

- Faster to evaluate than Gaussian
- More weight on tails
  - Spreads out low dimensional points



t-SNE algorithm

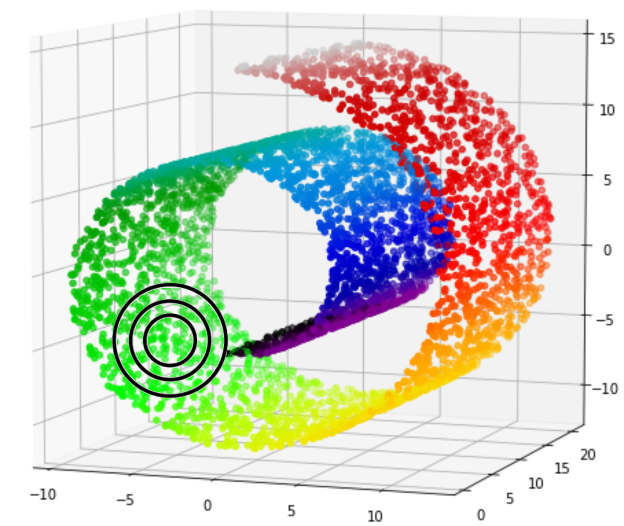
1. Calculate probability of selecting pair of points x_i, x_j

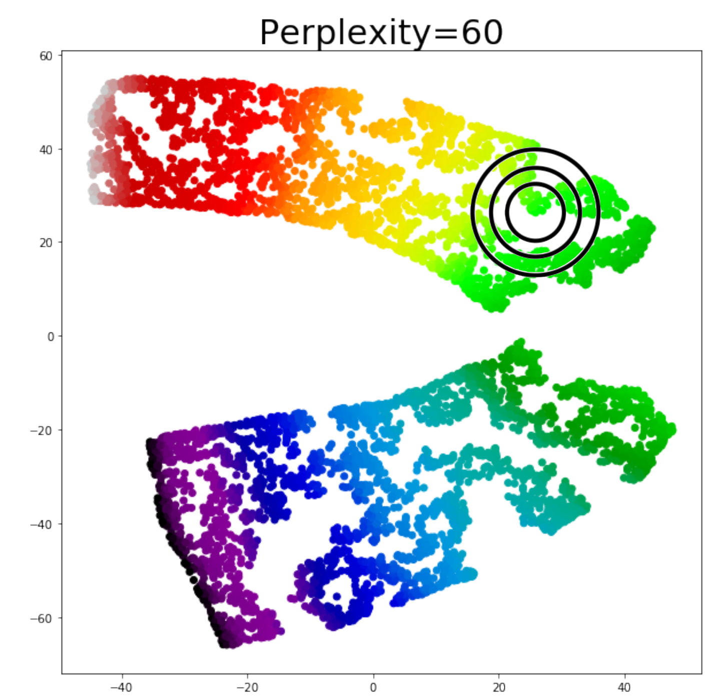
2. Initialise $y_1, \ldots, y_n$

3. Calculate probability of selecting pair of points y_i, y_j

4. Move $y_1, \ldots, y_n$ to decrease distance between probability distributions

5. Repeat 3-4 until convergence





van der Maaten & Hinton, 2008

t-SNE algorithm

1. Calculate probability of selecting pair of points x_i, x_j

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

$$p_{j|i} = \frac{\exp\left(- \parallel x_i - x_j \parallel^2 / 2\sigma_i^2\right)}{\sum_{l \neq i} \exp\left(- \parallel x_i - x_l \parallel^2 / 2\sigma_i^2\right)}$$

2. Initialise $y_1, \ldots, y_n$

3. Calculate probability of selecting pair of points y_i, y_j

$$q_{ij} = \frac{\left(1 + \parallel y_i - y_j \parallel^2\right)^{-1}}{\sum_{l \neq i}\left(1 + \parallel y_i - y_l \parallel^2\right)^{-1}}$$

4. Move $y_1, \ldots, y_n$ to decrease distance between probability distributions

$$KL(P \| Q) = \sum_i \sum_{j \neq 1} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

5. Repeat 3-4 until convergence

van der Maaten & Hinton, 2008

Hyperparameters

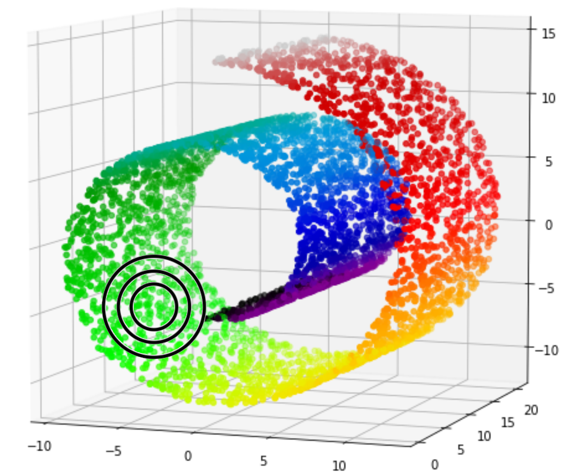
- Parameters for gradient descent
- Perplexity = smooth measure of number of neighbours
    - Number of neighbours roughly equal for each x_i
    - Sigma differs for each x_i

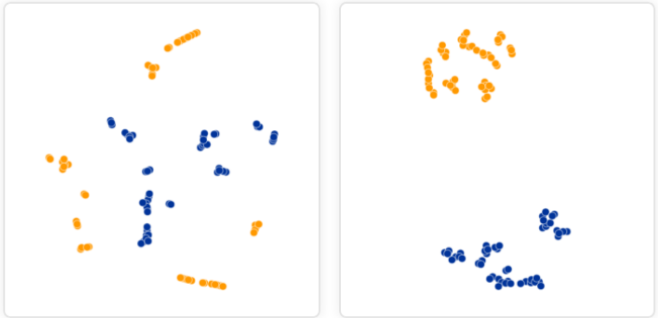
Perplexity: 2 Step: 5,000 | Perplexity: 5 Step: 5,000 | Perplexity: 30 Step: 5,000 | Perplexity: 50 Step: 5,000 | Perplexity: 100 Step: 5,000

Wattenberg, et al., 2016

t-distributed Stochastic Neighbour Embedding

- Preserves local similarity
- Stochastic
- Hyperparameters
    - Perplexity
- No explicit mapping



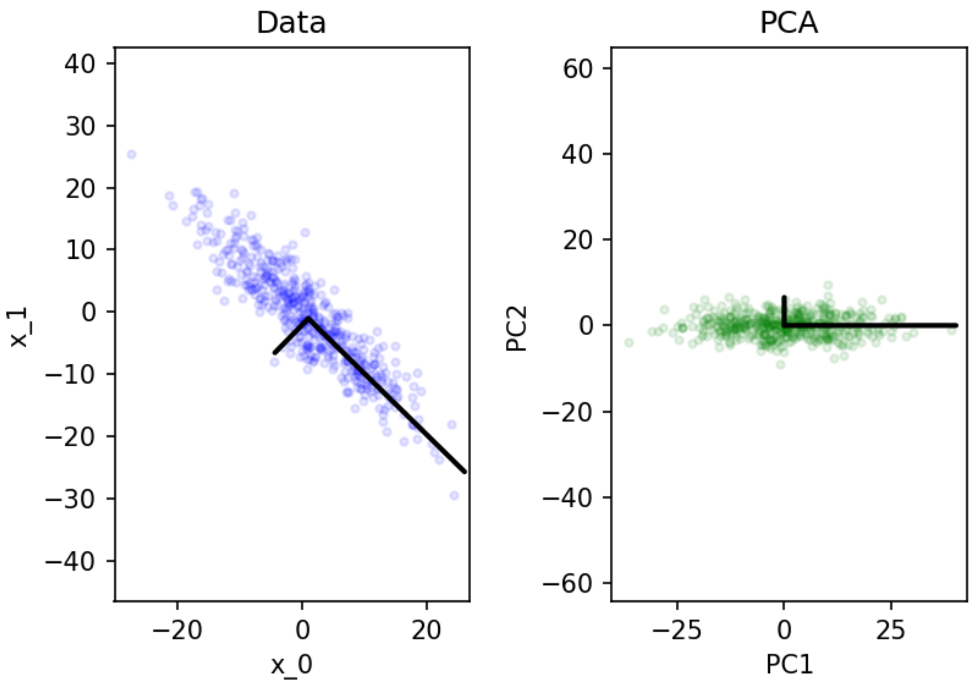


Perplexity: 2
Step: 5,000

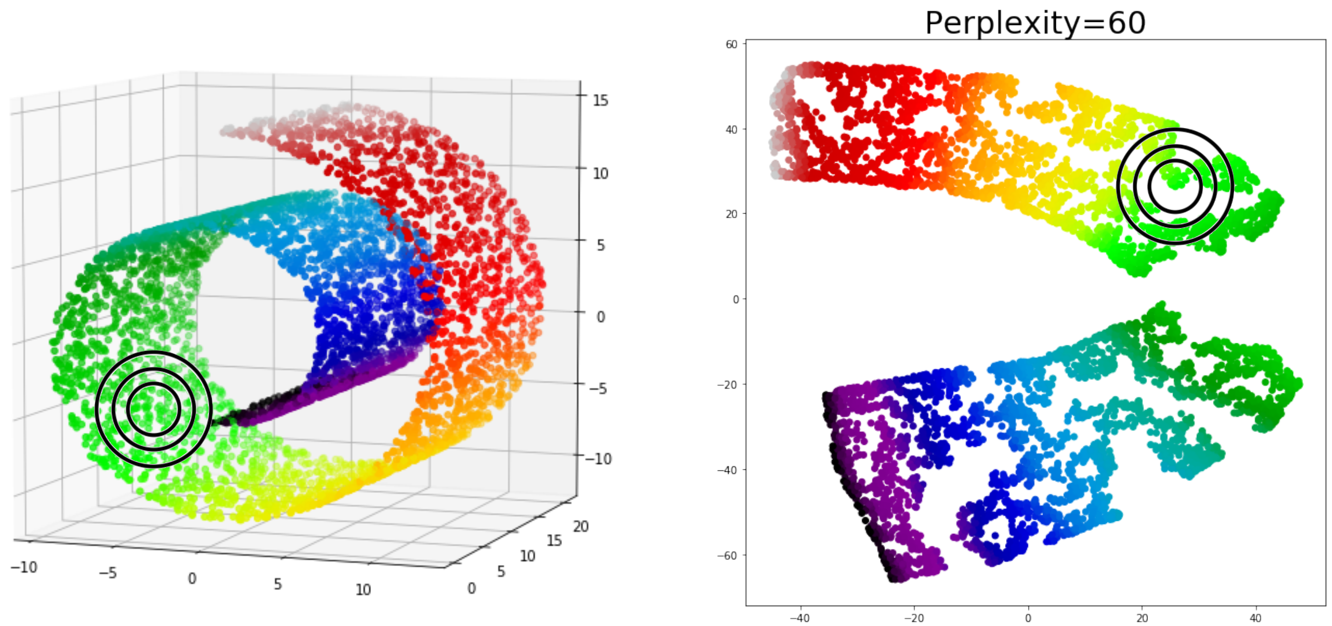
Perplexity: 5
Step: 5,000

Dimensionality reduction

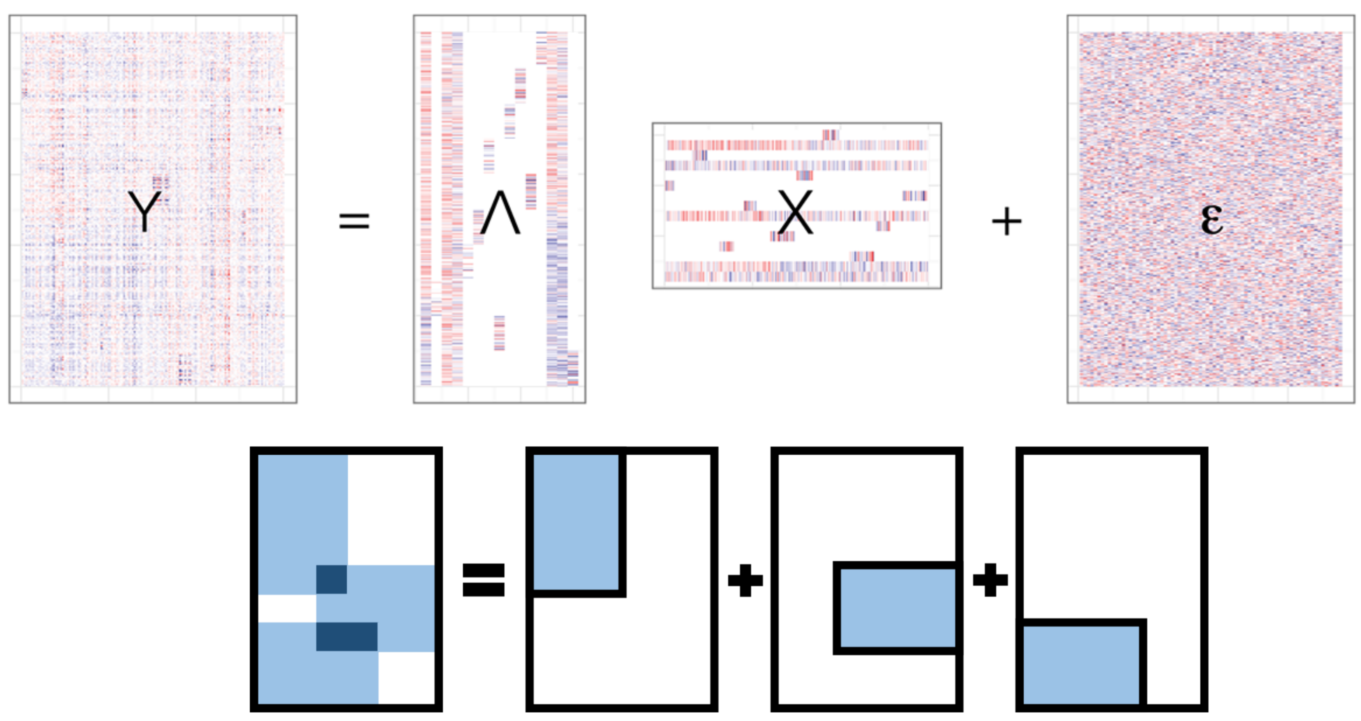
Principal Components Analysis



t-Stochastic Neighbour Embedding



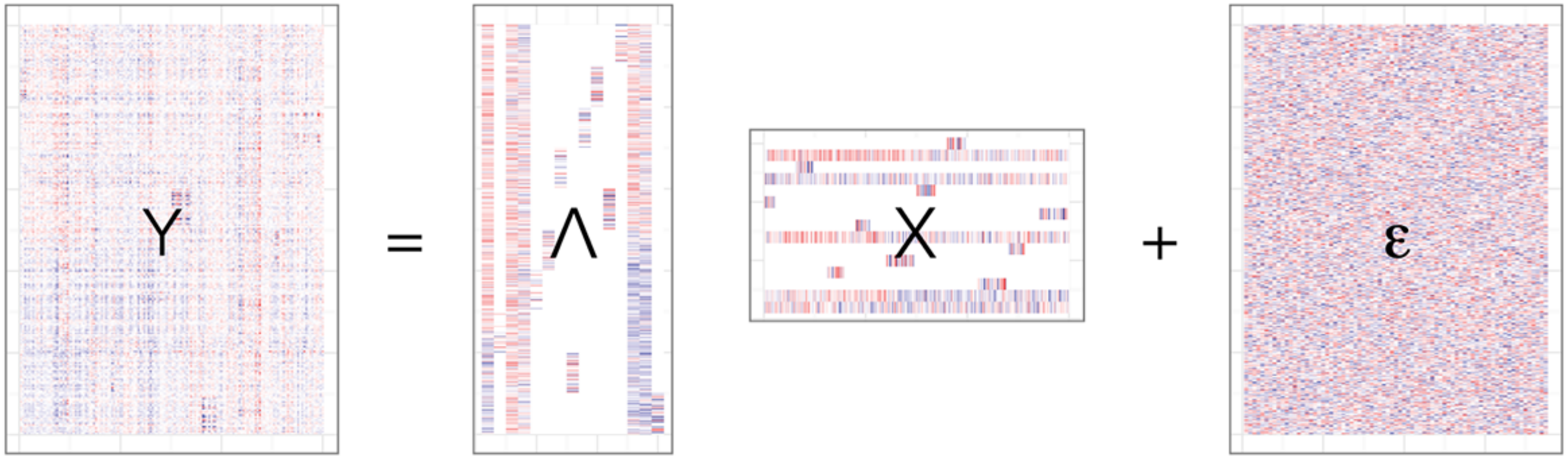
Sparse Factor Analysis



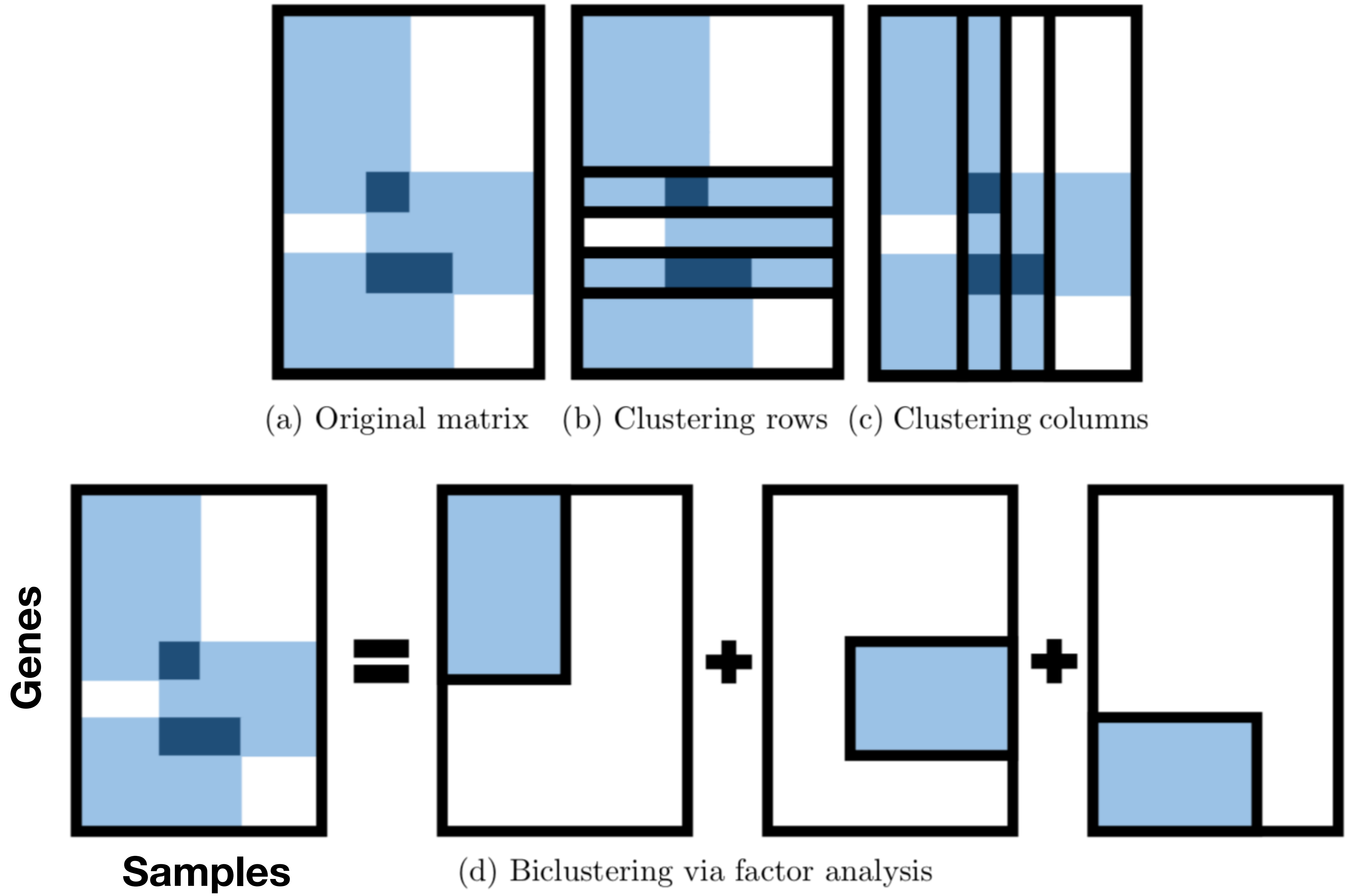
Kath Nicholls

Sparse Factor Analysis

- Identify latent variables in data
    - Explicit mapping to latent space
- Stochastic
- Sparsity helps choice of k



Sparse factor analysis finds biclusters

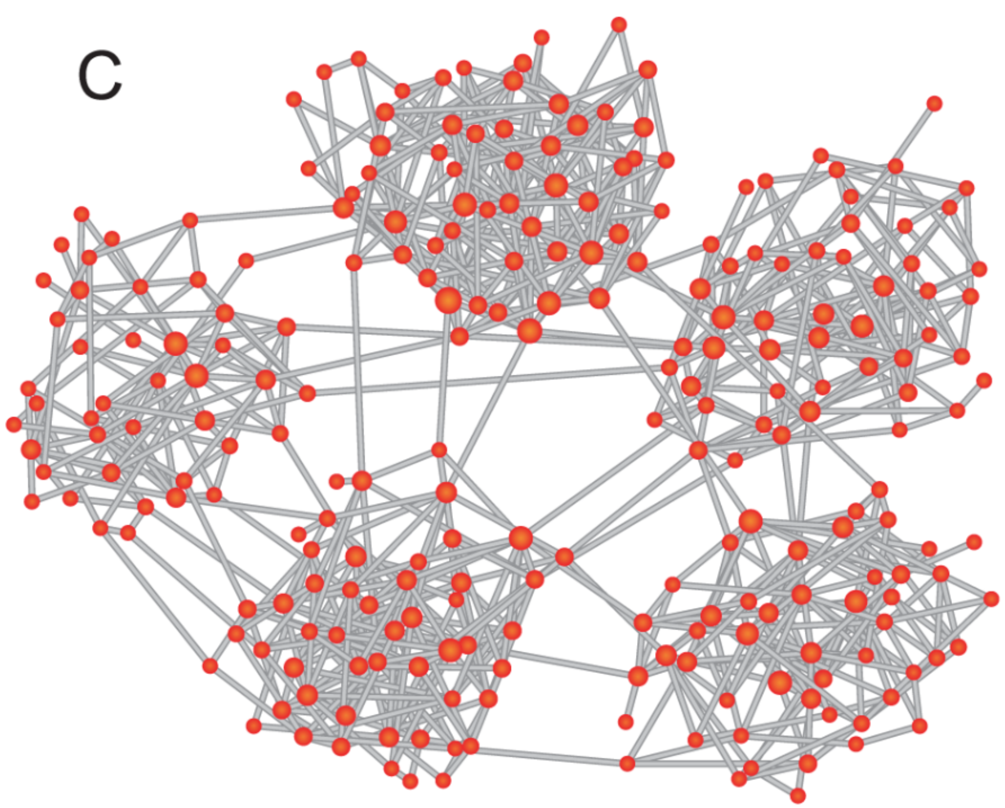


(a) Original matrix (b) Clustering rows (c) Clustering columns

(d) Biclustering via factor analysis

Aim of biclustering

- Gene regulatory networks are thought to be modular
- Biclustering finds modules, and identifies links between modules and disease (and/or cell type)



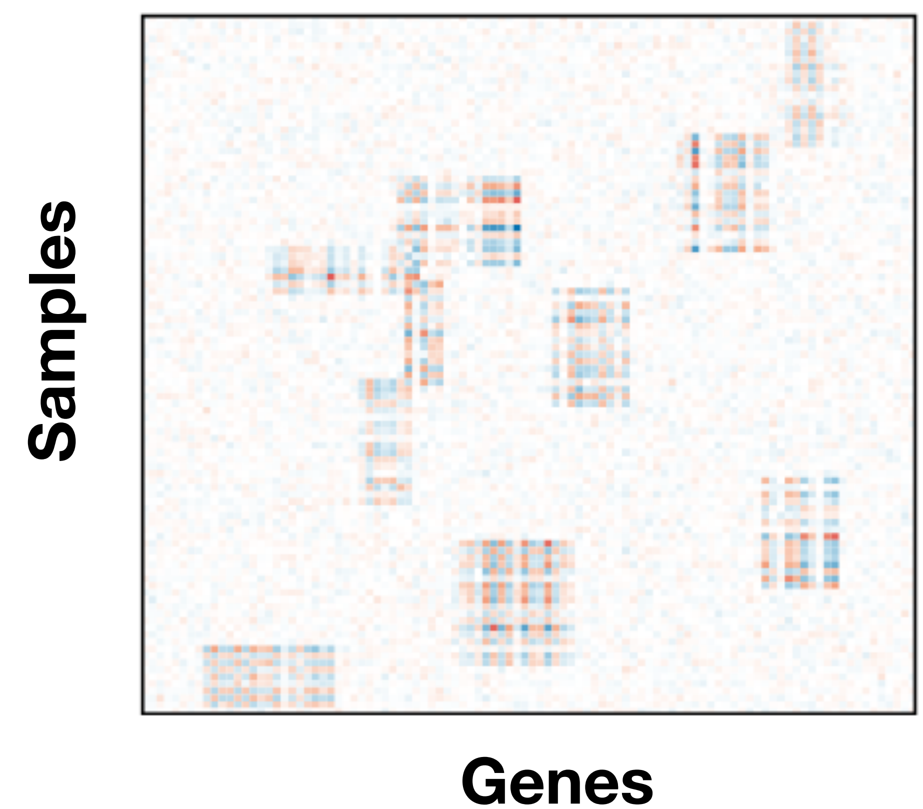


Figure 1C Müller-Linow et al. 2008; Figure 3 Moran et al. 2019

Comparison

PCA	**t-SNE**	**Sparse Factor Analysis**
Deterministic	Stochastic	Stochastic
Linear	**Non-linear**	Linear
Interpretable	Not interpretable	**Interpretable**
Mapping can be applied to other datasets	Mapping only works for one dataset	**Mapping can be applied to other datasets**
Easy to adjust for k	Have to re-run for each k	Have to re-run for each k
Good for explicit reduction	Good for visualisation	Good for biclusters